Analiza si vizualizarea datelor

Nicoleta ROGOVSCHI

nicoleta.rogovschi@parisdescartes.fr

Multi-Dimensional Scaling (MDS)

Outline

- Introduction and definitions
- Problem Formulation
- Algorithm
- Example
- Conclusions

Dimension reduction via feature extraction

Two main types of methods :

- **Linear Methods**
  - Principal Components Analysis (PCA)
  - Linear Discriminant Analysis (LDA)
  - Multi-Dimensional Scaling (MDS)
  - …
- **Non-Linear Methods**
  - Isometric feature mapping (Isomap)
  - Locally Linear Embedding (LLE)
  - Kernel PCA
  - Spectral clustering
  - Supervised methods (S-Isomap)
  - …

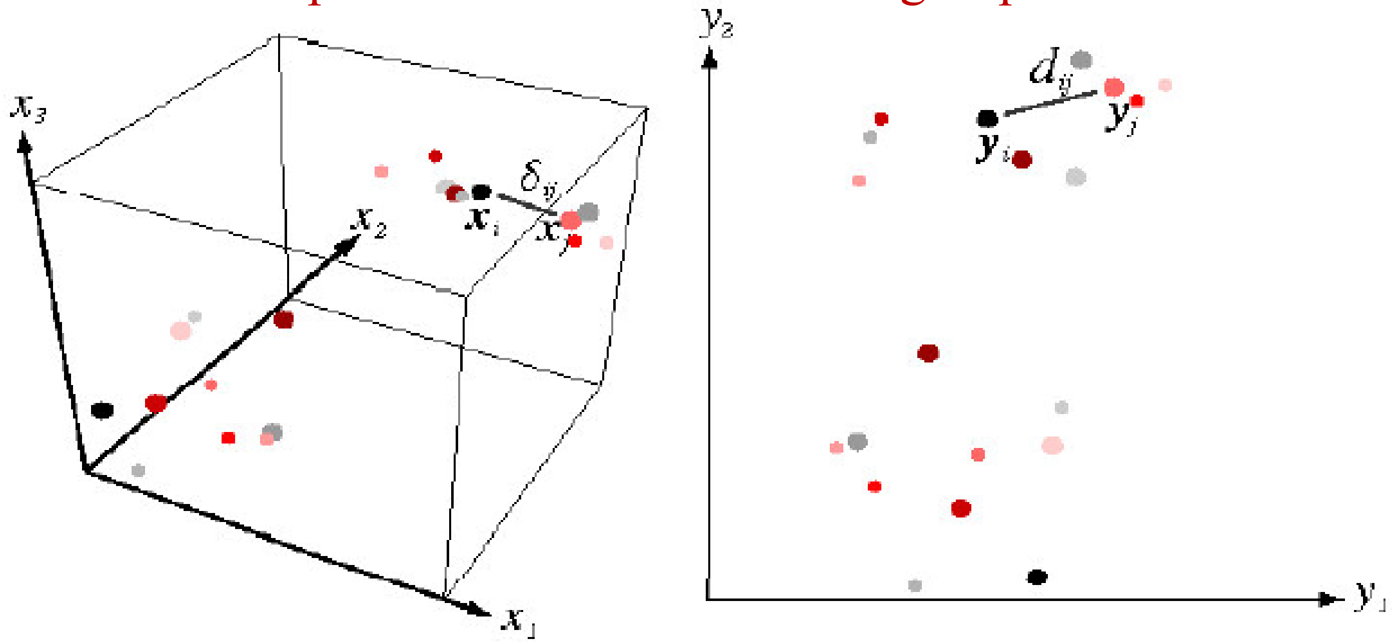
Introduction

- Multi-dimensional scaling (MDS) *(proposed by Borg and Groenen in 1997)*
  - A collection of dimension reduction techniques that maps the distances between observations in a high dimensional space into a lower dimensional space
  - Find a configuration of points in a low dimensional space whose inter-point distances correspond to dissimilarities in higher dimensions

Introduction

- Able to model intrinsic complex manifold structures and visualize them

Source space Target space



Multi-Dimensional Scaling (MDS)

In many applications :

- We known distances between the points of a data set
- We seek a representation in a low-dimensional space of these points

The method of multidimensional scaling (MDS) allows us to build this representation

Multi-Dimensional Scaling (MDS)

- **Example** :

➢ Get the map of a country starting from the knowledge of the distances between each pair of cities.

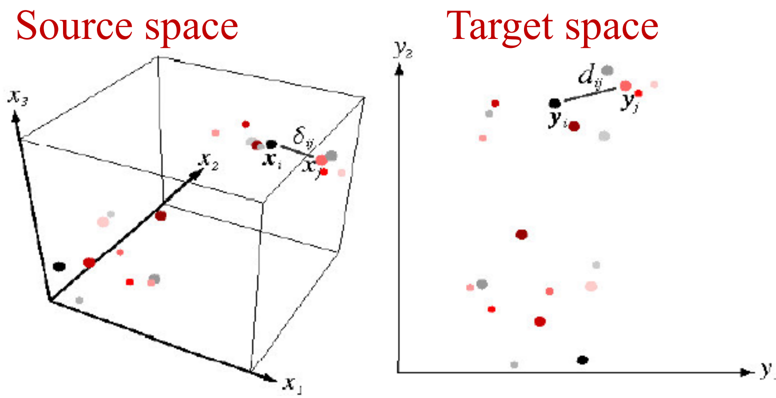
- As PCA, the MDS algorithm is based on the search of the eigenvalues.
- MDS builds a configuration of n points in R^d from N distances between objects.

Multi-Dimensional Scaling (MDS)

- So we have N(N-1)/2 distances. It is always possible to generate a position of N points in N dimensions that meets exactly the given distances.

- MDS computes an approximation in dimensions $d<N$.

Problem Formulation



- Given
  - The points $x_1,.., x_n$ in **k** dimensions
  - we note by δ_{ij} the distance between points x_i and x_j
- Find
  - The points $y_1,\ldots, y_n$ in 2 (or 3) dimensions, s.t. distance d_{ij} between y_i and y_j be close to δ_{ij}

Cost function

- We must search δ_{ij} which minimizes an objectif fonction
- We can define the cost function in a general maner:

$$Cost\_function = \sum_{i<j} (d_{ij} - \delta_{ij})^2$$

$$\delta_{ij} = \| x_i - x_j \|^2$$

$$d_{ij} = \| y_i - y_j \|^2$$

Examples of cost functions

- Possible Cost Functions (Stress)
  - d_{ij} is a function of y_i and y_j, and given the data the δ_{ij}'s are constant.

Disparity

$$J_{aa} = \frac{\sum_{i<j} (d_{ij} - \delta_{ij})^2}{\sum_{i<j} \delta_{ij}^2}$$ penalizes large absolute errors

$$J_{rr} = \sum_{i<j} \left(\frac{d_{ij} - \delta_{ij}}{\delta_{ij}} \right)^2$$ penalizes large relative errors

$$J_{ar} = \frac{1}{\sum_{i<j} \delta_{ij}} \sum_{i<j} \frac{(d_{ij} - \delta_{ij})^2}{\delta_{ij}}$$ a compromise between the two

Sammon Criterium

Update rules

- Update rules

$$\nabla J_{aa}(y_k) = \frac{2}{\sum_{i<j} \delta_{ij}^2} \sum_{j \neq k} (d_{kj} - \delta_{kj}) \frac{y_k - y_j}{d_{kj}}$$

$$\nabla J_{rr}(y_k) = 2 \sum_{j \neq k} \frac{d_{kj} - \delta_{kj}}{\delta_{kj}^2} \frac{y_k - y_j}{d_{kj}}$$

$$\nabla J_{ar}(y_k) = \frac{2}{\sum_{i<j} \delta_{ij}} \sum_{j \neq k} \frac{d_{kj} - \delta_{kj}}{\delta_{kj}} \frac{y_k - y_j}{d_{kj}}$$

Algorithm

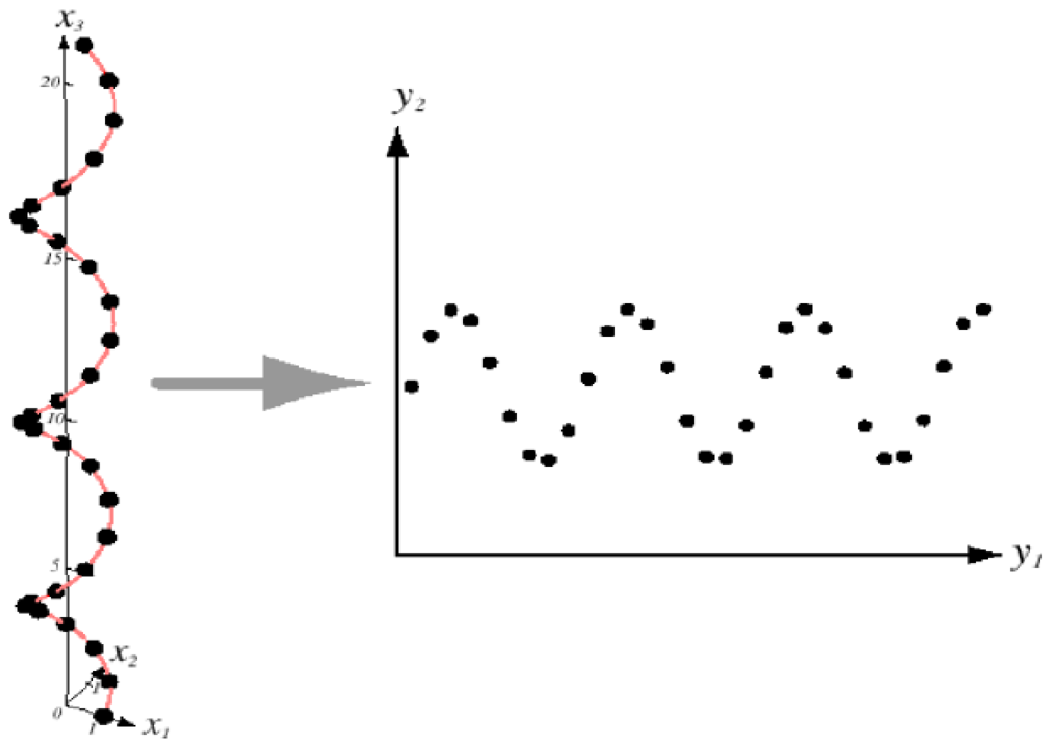
- Compute or obtain distances δ_{ij}
- Initialize the points $y_1, \ldots, y_n$ (e.g. randomly)
- Until convergence,

$$\forall i \quad y_i \leftarrow y_i - \eta \nabla J(y_i) \quad (0 < \eta < 1)$$

Example

- Artificial data set : we pass from a

3-dimensional space to 2-dimensionsal space



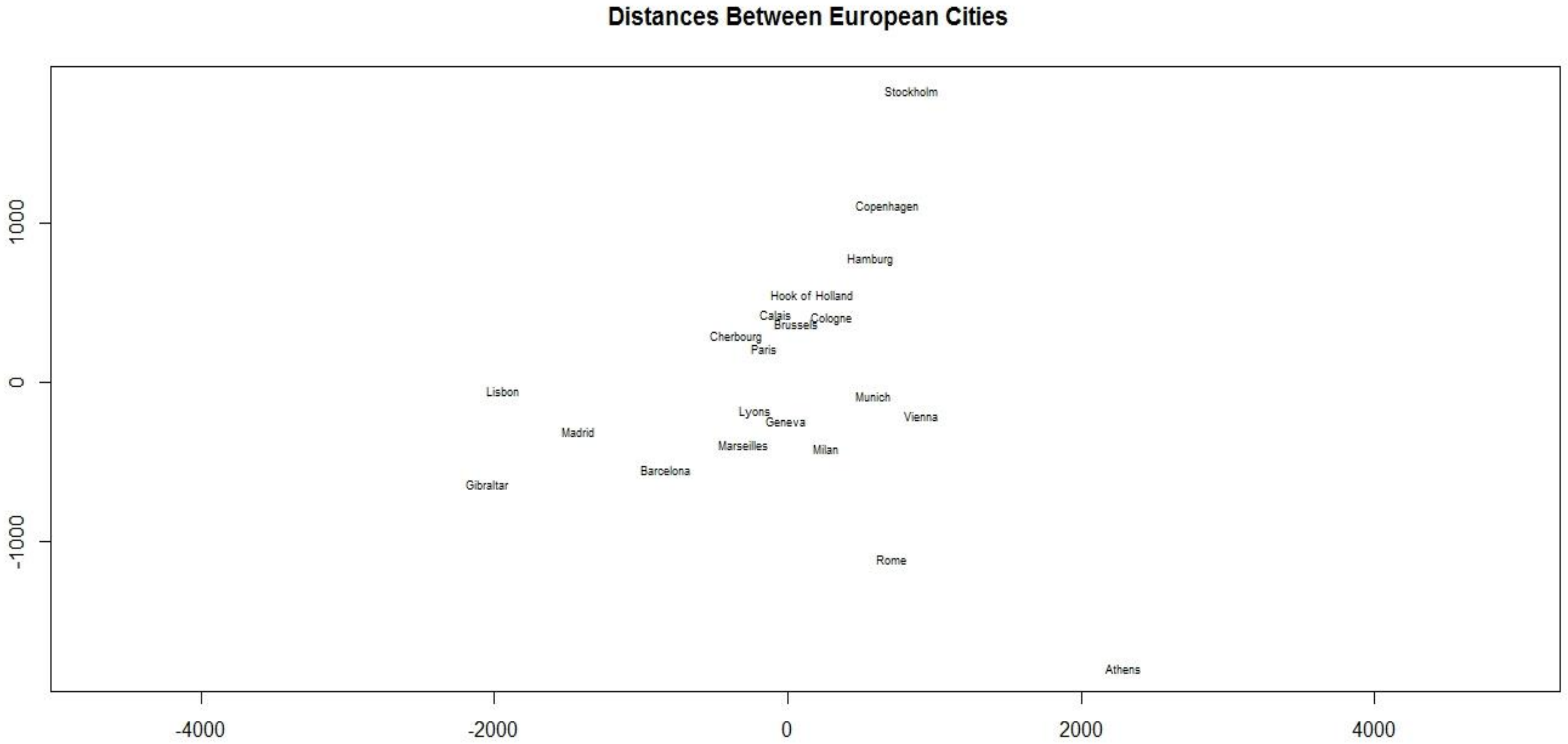
Example

- The data set "Eurodist" represents the distance (in km) between 21 cities of Europe.

- The source data set must be presented as a square matrix of dissimilarities between variables.

	Athens	Barcelona	Brussels	Calais	Cherbourg	Cologne	Copenhagen	Geneva	Gibraltar	Hamburg
Barcelona	3313									
Brussels	2963	1318								
Calais	3175	1326	204							
Cherbourg	3339	1294	583	460						
Cologne	2762	1498	206	409	785					
Copenhagen	3276	2218	966	1136	1545	760				
Geneva	2610	803	677	747	853	1662	1418			
Gibraltar	4485	1172	2256	2224	2047	2436	3196	1975		
Hamburg	2977	2018	597	714	1115	460	460	1118	2897	

Example


Distances Between European Cities
Stockholm
Copenhagen
Hamburg
Hook of Holland
Calais
Brussels
Cologne
Cherbourg
Paris
Lisbon
Munich
Lyons
Geneva
Vienna
Madrid
Marseilles
Milan
Barcelona
Gibraltar
Rome
Athens
1000
0
-1000
-4000
-2000
0
2000
4000

Conclusions

- MDS algorithms differ in :
  - The distance used in the source space
  - The Stress (objective) fonctions; the use of different stress functions leads to various results
  - The optimization procedure ; linear MDS has analytic solvable but cannot model complex (nonlinear) low-dimensional manifold well while nonlinear MDS often needs to use an iterative algorithm